

Mango-AM335x-ST 보드 UART 6 Port 사용하기

<http://www.mangoboard.com/>

<http://cafe.naver.com/embeddedcrazyboys>

Crazy Embedded Laboratory

Document History

Revision	Date	Change note
Init	2015-08-10	전종인

1. UART 6 Port 사용하기	4
1.1. Mango-AM335x-ST UART 1 검증	5
1.1.1. 소스 수정 및 확인	5
1.1.2. UART1 테스트	7
1.2. Mango-AM335x-ST 보드 UART2 검증	10
1.2.3. 소스 수정 및 확인	11
1.2.4. UART2 테스트	14
1.3. Mango-AM335x-ST 보드 UART3검증	15
1.3.5. 소스 수정 및 확인	16
1.3.6. Uart3 테스트	18
1.4. Mango-AM335x-ST 보드 UART4검증	20
1.4.7. 소스 수정	21
1.4.8. UART4 테스트	23
1.5. Mango-AM335x-ST 보드 UART5검증	25
1.5.9. Uart5 검증을 위한 소스 수정	27
1.5.10. UART5 TX , RX 테스트	30

1. UART 6 Port 사용하기

AM335x는 6개 Port를 지원합니다.

Mango-AM335x-ST 보드에서 UART를 사용할 수 있도록 하려면 작업이 필요합니다.

UART0, UART1 은 확장 포트에서 이 이름으로 바로 사용 가능

나머지 UART 는 확장 커넥터 J12 번에서

-UART2 는 GPIO0_2 : UART2_RXD, GPIO0_3 : UART2_TXD

-UART3 는 GPIO0_6 : UART3_RXD, GPIO0_7 : UART3_TXD

-UART4 는 I2C1_SDA : UART4_RXD, I2C1_SCL : UART4_TXD 다만 이 두 신호선은 1.5K 저항으로 3.3V 로 풀업 되어 있음에 주의

UART 로 사용하는데는 아무 문제 없음

-UART5 는 GPIO3_0 : UART5_RXD, GPIO0_29 : UART5_TXD

보드 배치도를 살펴 보겠습니다.

J13			
VDD_3.3V	1	2	NC
UART1_TXD	3	4	NC
UART1_RXD	5	6	NC
NC	7	8	NC
NC	9	10	NC
GPIO3_0	11	12	NC
GND	13	14	NC
USBHUB_DM4	15	16	UART1_CTSN
USBHUB_DP4	17	18	UART1_RTSN
GND	19	20	NC

J12			
VDD_3.3V	1	2	GPIO3_14
VDD_3.3V	3	4	GPIO3_15
GPIO0_2(UART2_RXD)	5	6	GPIO3_16
GPIO0_3(UART2_TXD)	7	8	GPIO3_17
GPIO0_4	9	10	NC
GPIO0_5	11	12	NC
GPIO0_6(UART3_RXD)	13	14	GPIO3_20
GND	15	16	GPIO3_21
UART0_TXD	17	18	GND
UART0_RXD	19	20	GPIO3_7(JTAG_EMU0)
I2C1_SCL(UART4_TXD)	21	22	GPIO3_8(JTAG_EMU1)
I2C1_SDA(UART4_RXD)	23	24	GPIO3_1
UART1_TXD	25	26	GPIO3_0(UART5_RXD)
UART1_RXD	27	28	GPIO3_2
UART1_CTSN	29	30	GPIO0_29(UART5_TXD)
UART1_RTSN	31	32	GPIO0_7(UART3_TXD)
GND	33	34	GND
USBHUB_DM3	35	36	NC
USBHUB_DP3	37	38	VDD_3v3
VDD_5v0	39	40	VDD_3v3

DEBUG UART0

UART 0번은 Debug용으로 사용하므로 이미 검증이 되었습니다.

UART 1 ~ 5번까지 소프트웨어 작업을 하고 , 검증을 해 보도록 하겠습니다.

소프트웨어 작업은 아래 링크 참조하여 작업을 하면 됩니다.

https://e2e.ti.com/support/arm/sitara_arm/f/791/t/262312

1.1. Mango-AM335x-ST UART 1 검증

J13			
VDD_3.3V	1	2	NC
UART1_TXD	3	4	NC
UART1_RXD	5	6	NC
NC	7	8	NC
NC	9	10	NC
GPIO3_0	11	12	NC
GND	13	14	NC
USBHUB_DM4	15	16	UART1_CTSN
USBHUB_DP4	17	18	UART1_RTSN
GND	19	20	NC

J12			
VDD_3.3V	1	2	GPIO3_14
VDD_3.3V	3	4	GPIO3_15
GPIO0_2(UART2_RXD)	5	6	GPIO3_16
GPIO0_3(UART2_TXD)	7	8	GPIO3_17
GPIO0_4	9	10	NC
GPIO0_5	11	12	NC
GPIO0_6(UART3_RXD)	13	14	GPIO3_20
GND	15	16	GPIO3_21
UART0_TXD	17	18	GND
UART0_RXD	19	20	GPIO3_7(UTAG_EMU0)
I2C1_SCL(UART4_TXD)	21	22	GPIO3_8(UTAG_EMU1)
I2C1_SDA(UART4_RXD)	23	24	GPIO3_1
UART1_TXD	25	26	GPIO3_0(UART5_RXD)
UART1_RXD	27	28	GPIO3_2
UART1_CTSN	29	30	GPIO0_29(UART5_TXD)
UART1_RTSN	31	32	GPIO0_7(UART3_TXD)
GND	33	34	GND
USBHUB_DM3	35	36	NC
USBHUB_DP3	37	38	VDD_3v3
VDD_5v0	39	40	VDD_3v3

DEBUG UART0

J13, J12 커넥터에 UART1 RX, TX, GND, VDD_3v3을 연결한 후 검증 했습니다.

1.1.1. 소스 수정 및 확인

커널에서 수정사항은 아래와 같습니다.

“arch/arm/mach-omap2/board-am335xevm.c”

```
static struct pinmux_config uart1_pin_mux[] = {
    {"uart1_ctsn.uart1_ctsn", OMAP_MUX_MODE0 | AM33XX_PIN_INPUT},
    {"uart1_rtsn.uart1_rtsn", OMAP_MUX_MODE0 | AM33XX_PIN_OUTPUT},
    {"uart1_rxd.uart1_rxd", OMAP_MUX_MODE0 | AM33XX_PIN_INPUT_PULLUP},
    {"uart1_txd.uart1_txd", OMAP_MUX_MODE0 | AM33XX_PULL_ENBL},
    {NULL, 0},
};
```

```
static void uart1_init(int evm_id, int profile)
{
    MANGO_DBG_DEFAULT;
```

```

setup_pin_mux(uart1_pin_mux);
return;
}

```

```

static struct evm_dev_cfg evm_sk_dev_cfg[] = {
..
{uart1_init,          DEV_ON_BASEBOARD, PROFILE_ALL},

```

"arch/arm/mach-omap2/mux33xx.c"

```

_AM33XX_MUXENTRY(UART1_RXD, 0,
    "uart1_rxd", "mmc1_sdwp", NULL, "i2c1_sda",
    NULL, "pr1_uart0_rxd_mux1", NULL, "gpio0_14"),
_AM33XX_MUXENTRY(UART1_TXD, 0,
    "uart1_txd", "mmc2_sdwp", NULL, "i2c1_scl",
    NULL, "pr1_uart0_txd_mux1", NULL, "gpio0_15"),

```

"arch/arm/mach-omap2/omap_hwmod_33xx_data.c" 에 아래와 같이 정의가 되어 있어야 합니다.

```

/* uart1 */
static struct omap_hwmod_dma_info uart1_edma_reqs[] = {
    { .name = "tx",    .dma_req = 26, },
    { .name = "rx",    .dma_req = 27, },
    { .dma_req = -1 }
};

static struct omap_hwmod_addr_space am33xx_uart1_addr_space[] = {
    {
        .pa_start = 0x44E09000,
        .pa_end   = 0x44E09000 + SZ_8K - 1,
        .flags    = ADDR_TYPE_RT,
    },
    {}
};

static struct omap_hwmod_ocp_if am33xx_l4_wkup_uart1 = {

```

```

        .master          = &am33xx_l4wkup_hwmod,
        .slave          = &am33xx_uart1_hwmod,
        .clk            = "uart1_ick",
        .addr           = am33xx_uart1_addr_space,
        .user           = OCP_USER_MPU,
};

static struct omap_hwmod_irq_info am33xx_uart1_irqs[] = {
    { .irq = 72 },
    { .irq = -1 }
};

static struct omap_hwmod_ocp_if *am33xx_uart1_slaves[] = {
    &am33xx_l4_wkup__uart1,
};

static struct omap_hwmod am33xx_uart1_hwmod = {
    .name              = "uart1",
    .class             = &uart_class,
    .clkdm_name       = "l4_wkup_clkdm",
    .mpu_irqs         = am33xx_uart1_irqs,
    .main_clk         = "uart1_fck",
    .sdma_reqs        = uart1_edma_reqs,
    .prcm             = {
        .omap4         = {
            .clkctrl_offs = AM33XX_CM_WKUP_UART0_CLKCTRL_OFFSET,
            .modulemode  = MODULEMODE_SWCTRL,
        },
    },
    .slaves           = am33xx_uart1_slaves,
    .slaves_cnt       = ARRAY_SIZE(am33xx_uart1_slaves),
};

```

1.1.2. UART1 테스트

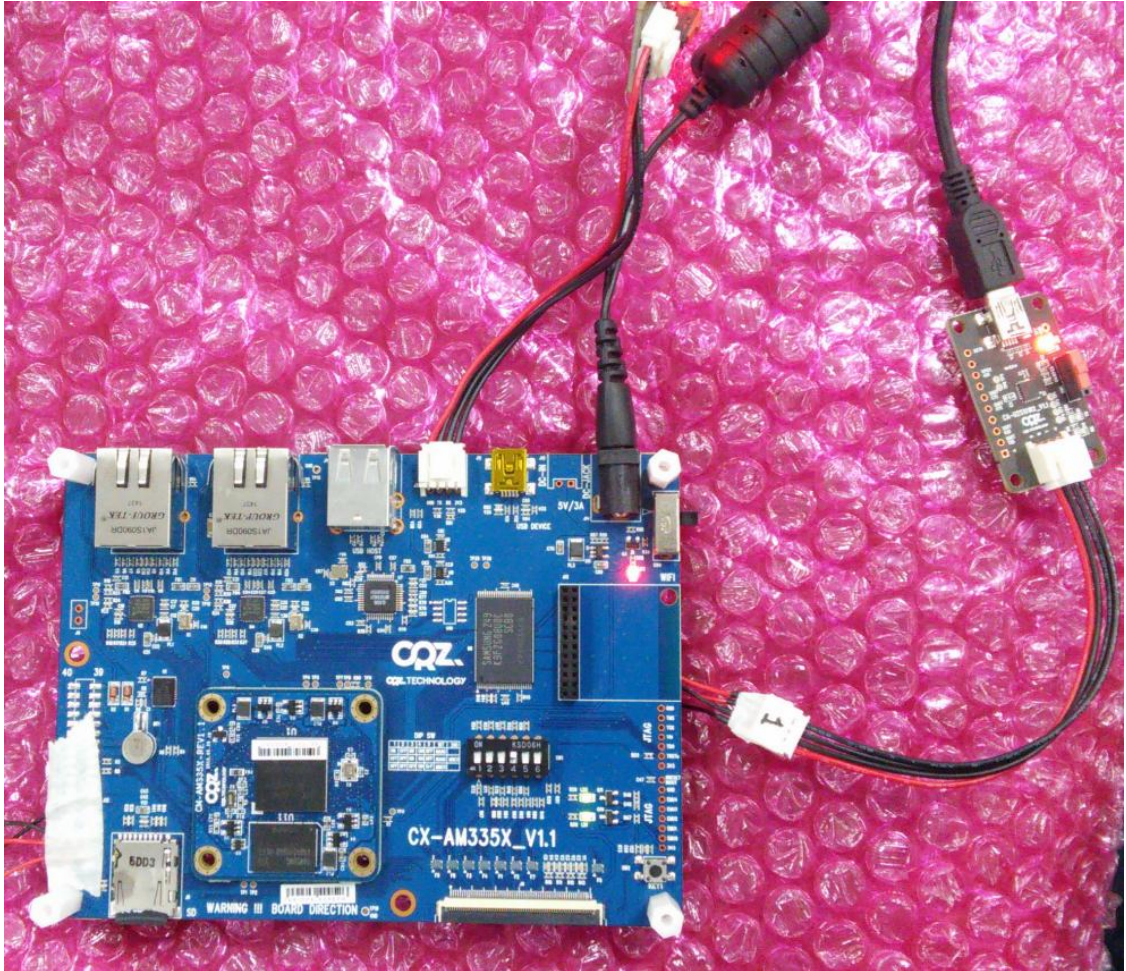
테스트 방법

부팅 후 디버깅 터미널에서

```
# stty -F /dev/ttyO1 115200
```

Baudrate를 115200으로 설정합니다.

보드에 usb to serial uart 보드를 2개 연결합니다.



또 다른 터미널 프로그램을 실행을 합니다. Baudrate는 115200으로 설정합니다.

TX 테스트

```
# ls -al > /dev/ttyO1
```

디버깅 터미널 창에서 위와 같이 입력을 하면

UART1 이 연결된 터미널 창으로 아래와 같이 출력이 됩니다.


```

total 144
drwxrwxr-x  5 1001 1001  4096 Apr 20  2015 ./
drwxrwxr-x 18 root root   4096 Aug 10  2015 ../
-rw-rw-r--  1 root root     0 Nov 30  2013 .bash_history
-rw-rw-r--  1 root root   175 Nov 30  2013 .bash_logout
-rw-rw-r--  1 root root   161 Nov 30  2013 .bash_profile
-rwxr--r--  1 1001 1001  1019 Apr 20  2015 S99_total_mp_process.sh*
drwxrwxr-x  2 1001 1001  4096 Feb 24  2015 boot_dir/
-rwxr--r--  1 1001 1001   150 Feb 24  2015 flash_eraseall*
-rw-rw-r--  1 1001 1001 73805 May 19  2015 g_mass_storage.ko
drwxrwxr-x  2 1001 1001  4096 Feb  5  2014 nfs_mount/
-rwxr-xr-x  1 1001 1001   233 Feb  5  2014 nfs_mount.sh*
-rwxrwxr-x  1 1001 1001 10391 Mar 21  2014 serial_test*
-rwxr-xr-x  1 1001 1001   196 Feb 24  2015 tmp_kernel_update.sh*
-rwxr-xr-x  1 1001 1001   344 Mar  9  2015 tmp_uboot_update.sh*
drwxr-xr-x  2 1001 1001  4096 Feb 25  2015 total_mp_process/
-rwxr-xr-x  1 1001 1001   154 Apr 23  2014 usb_dev_mass.sh*

```

RX 테스트

```
# cat /dev/ttyO1
```

디버깅 창에서 위와 같이 입력합니다.

다른 UART1 port가 연결 된 터미널 프로그램에서 아무 글자를 입력하면 그대로 출력이 됩니다.

```
[root@localhost ~]# cat /dev/ttyO1
```

```
dfasdfasdafdf
```

```
d
```

```
fa
```

```
sdaf
```

```
sd
```

```
fa
```

```
df
```

```
a
```

```
sdf
```

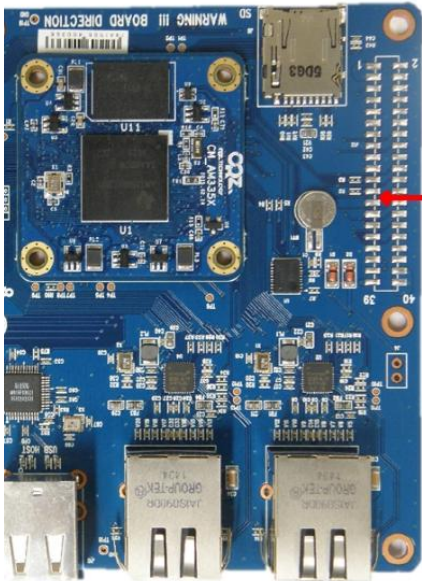
```
ads
```

```
fa
```

```
sdf
```

a
dfasdfasdf
adsf

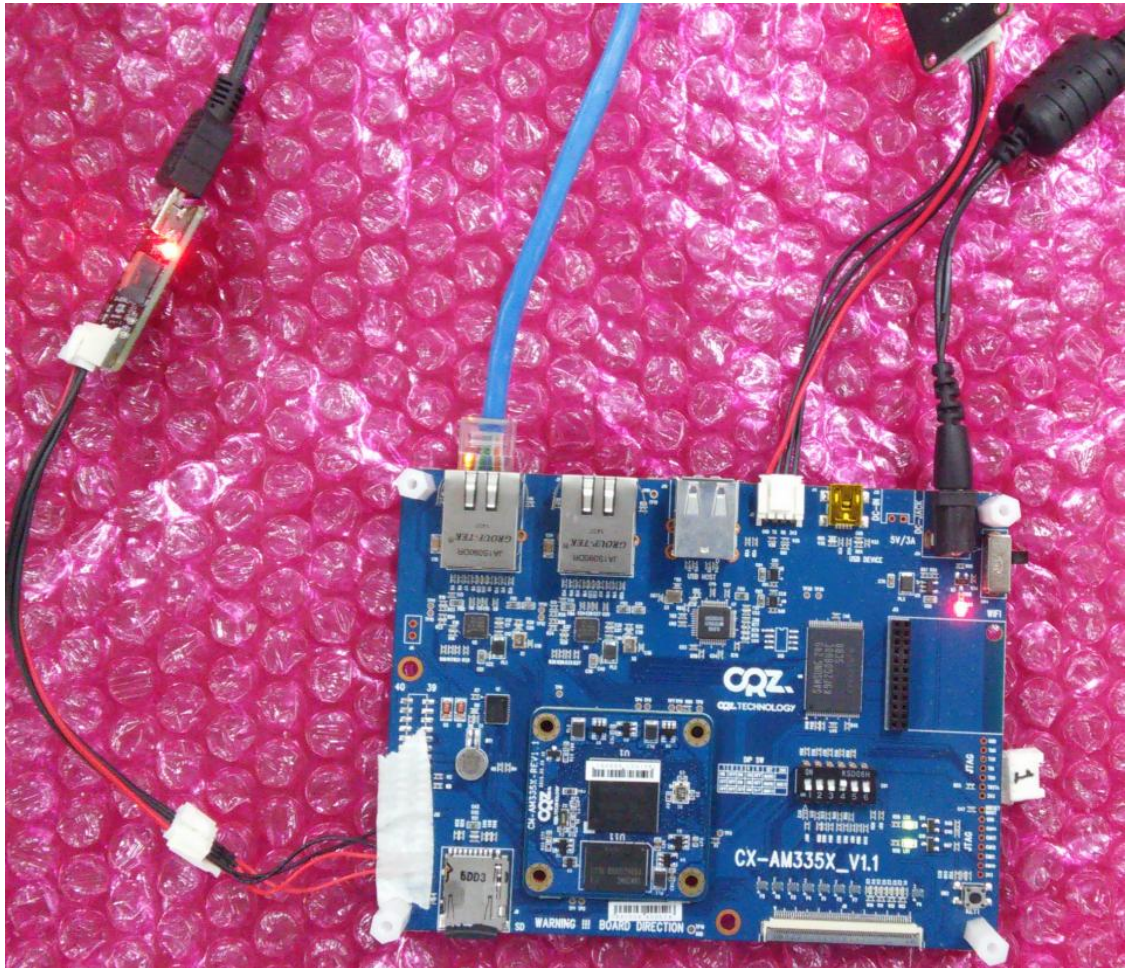
1.2. Mango-AM335x-ST 보드 UART2 검증



J12			
VDD_3.3V	1	2	GPIO3_14
VDD_3.3V	3	4	GPIO3_15
GPIO0_2(UART2_RXD)	5	6	GPIO3_16
GPIO0_3(UART2_TXD)	7	8	GPIO3_17
GPIO0_4	9	10	NC
GPIO0_5	11	12	NC
GPIO0_6(UART3_RXD)	13	14	GPIO3_20
GND	15	16	GPIO3_21
UART0_TXD	17	18	GND
UART0_RXD	19	20	GPIO3_7(JTAG_EMU0)
I2C1_SCL(UART4_TXD)	21	22	GPIO3_8(JTAG_EMU1)
I2C1_SDA(UART4_RXD)	23	24	GPIO3_1
UART1_TXD	25	26	GPIO3_0(UART5_RXD)
UART1_RXD	27	28	GPIO3_2
UART1_CTSN	29	30	GPIO0_29(UART5_TXD)
UART1_RTSN	31	32	GPIO0_7(UART3_TXD)
GND	33	34	GND
USBHUB_DM3	35	36	NC
USBHUB_DP3	37	38	VDD_3v3
VDD_5v0	39	40	VDD_3v3

J12 커넥터에서 UART2 RX, TX, GND, VDD_3v3을 연결 했습니다.

보드 UART2 (J12) <-> USB to Serial UART 보드 <-> PC 연결 했습니다.



1.2.3. 소스 수정 및 확인

소프트 수정 작업은

"arch/arm/mach-omap2/board-am335xevm.c"

```
static struct pinmux_config uart2_pin_mux[] = {  
    {"spi0_sclk. uart2_rxd", OMAP_MUX_MODE1 | AM33XX_SLEWCTRL_SLOW |  
                                     AM33XX_PIN_INPUT_PULLUP},  
    {"spi0_d0. uart2_txd", OMAP_MUX_MODE1 | AM33XX_PULL_UP |
```

```

                                AM33XX_PULL_DISA |
                                AM33XX_SLEWCTRL_SLOW},
    {NULL, 0},
};

```

```

static void uart2_init(int evm_id, int profile)
{
    MANGO_DBG_DEFAULT;

    setup_pin_mux(uart2_pin_mux);
    return;
}

```

```

static struct evm_dev_cfg evm_sk_dev_cfg[] = {
...
{uart2_init,          DEV_ON_BASEBOARD, PROFILE_ALL},

```

위와 같이 코드가 적용이 되어 있는지 확인 합니다.

"arch/arm/mach-omap2/mux33xx.c"

```

    _AM33XX_MUXENTRY(SPI0_SCLK, 0,
        "spi0_sclk", "uart2_rxd", "i2c2_sda", NULL,
        NULL, NULL, NULL, "gpio0_2"),
    _AM33XX_MUXENTRY(SPI0_D0, 0,
        "spi0_d0", "uart2_txd", "i2c2_scl", NULL,
        NULL, NULL, NULL, "gpio0_3"),

```

"arch/arm/mach-omap2/omap_hwmod_33xx_data.c"

```

/* uart2 */
static struct omap_hwmod_addr_space am33xx_uart2_addr_space[] = {
    {
        .pa_start = 0x48022000,
        .pa_end   = 0x48022000 + SZ_8K - 1,
        .flags    = ADDR_TYPE_RT,
    },

```

```

    {}
};

static struct omap_hwmod_ocp_if am33xx_l4_ls_uart2 = {
    .slave          = &am33xx_uart2_hwmod,
    .clk            = "uart2_ick",
    .addr           = am33xx_uart2_addr_space,
    .user           = OCP_USER_MPU,
};

static struct omap_hwmod_irq_info am33xx_uart2_irqs[] = {
    { .irq = 73 },
    { .irq = -1 }
};

static struct omap_hwmod_ocp_if *am33xx_uart2_slaves[] = {
    &am33xx_l4_ls_uart2,
};

static struct omap_hwmod am33xx_uart2_hwmod = {
    .name           = "uart2",
    .class          = &uart_class,
    .clkdm_name     = "l4ls_clkdm",
    .mpu_irqs       = am33xx_uart2_irqs,
    .main_clk       = "uart2_fck",
    .sdma_reqs      = uart1_edma_reqs,
    .prcm           = {
        .omap4      = {
            .clkctrl_offs = AM33XX_CM_PER_UART1_CLKCTRL_OFFSET,
            .modulemode  = MODULEMODE_SWCTRL,
        },
    },
    .slaves         = am33xx_uart2_slaves,
    .slaves_cnt     = ARRAY_SIZE(am33xx_uart2_slaves),
};

```

1.2.4. UART2 테스트

UART2 채널 테스트

```
# stty -F /dev/ttyO2 115200
```

Baudrate : 115200 설정합니다.

TX 테스트

```
# ls -al > /dev/ttyO2
```

위와 같이 명령을 입력하면 다른 터미널 창에서 출력이 됩니다.

```
total 144
drwxrwxr-x  5 1001 1001  4096 Apr 20  2015 ./
drwxrwxr-x 18 root root  4096 Aug 10  2015 ../
-rw-rw-r--  1 root root    0 Nov 30  2013 .bash_history
-rw-rw-r--  1 root root  175 Nov 30  2013 .bash_logout
-rw-rw-r--  1 root root  161 Nov 30  2013 .bash_profile
-rwxr--r--  1 1001 1001  1019 Apr 20  2015 S99_total_mp_process.sh*
drwxrwxr-x  2 1001 1001  4096 Feb 24  2015 boot_dir/
-rwxr--r--  1 1001 1001   150 Feb 24  2015 flash_eraseall*
-rw-rw-r--  1 1001 1001 73805 May 19  2015 g_mass_storage.ko
drwxrwxr-x  2 1001 1001  4096 Feb  5  2014 nfs_mount/
-rwxr-xr-x  1 1001 1001   247 Jan 16 17:16 nfs_mount.sh*
-rwxrwxr-x  1 1001 1001 10391 Mar 21  2014 serial_test*
-rwxr-xr-x  1 1001 1001   196 Feb 24  2015 tmp_kernel_update.sh*
-rwxr-xr-x  1 1001 1001   344 Mar  9  2015 tmp_uboot_update.sh*
drwxr-xr-x  2 1001 1001  4096 Feb 25  2015 total_mp_process/
-rwxr-xr-x  1 1001 1001   154 Apr 23  2014 usb_dev_mass.sh*
```

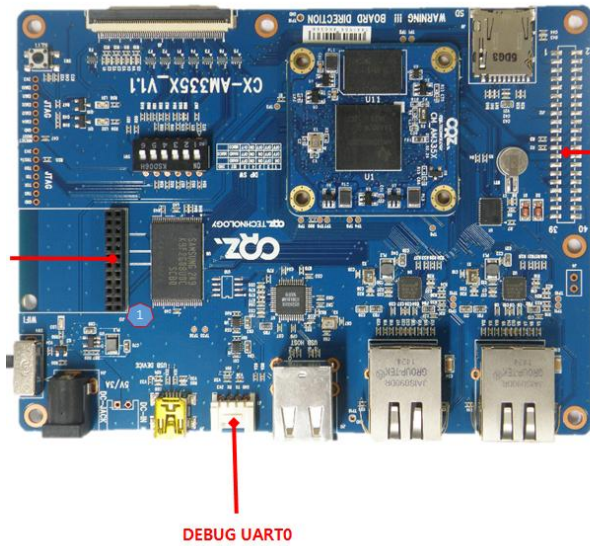
RX 테스트

디버깅 터미널 창에서 아래와 같이 입력하고, UART2가 연결 된 터미널 창에서 아래와 같이 입력하면 디버깅 터미널 창으로 출력이 됩니다.


```
[root@localhost ~]# cat /dev/ttyO2
```

Mango UART2 Test OK

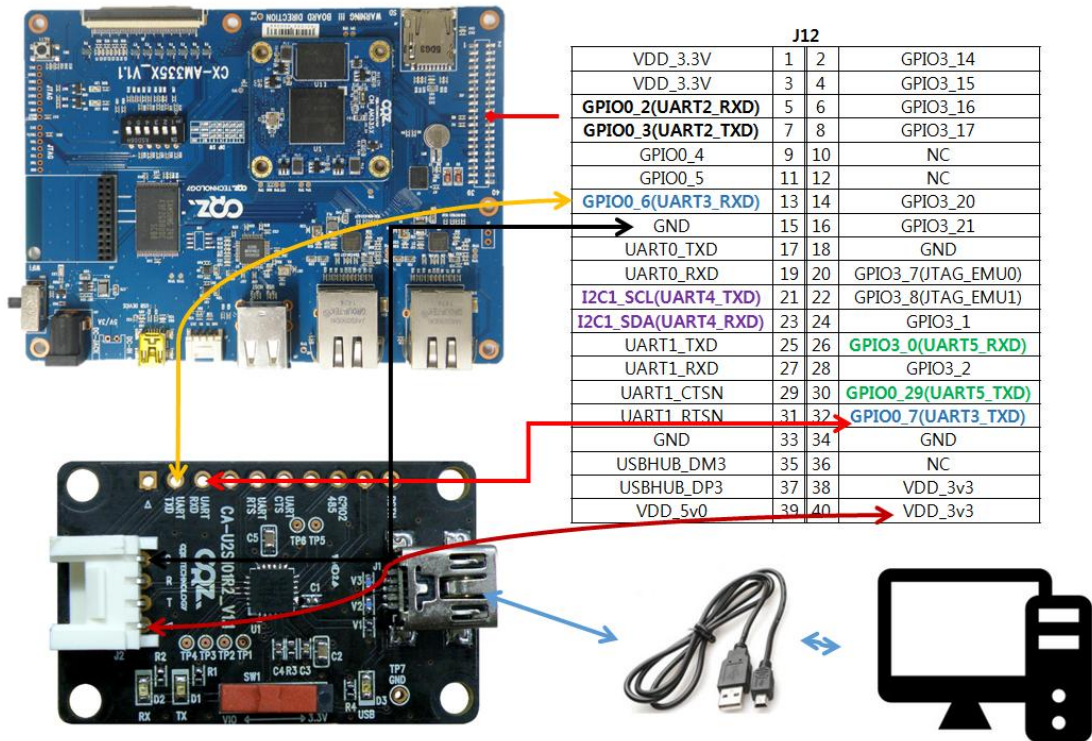
1.3. Mango-AM335x-ST 보드 UART3검증



J12			
VDD_3.3V	1	2	GPIO3_14
VDD_3.3V	3	4	GPIO3_15
GPIO0_2(UART2_RXD)	5	6	GPIO3_16
GPIO0_3(UART2_TXD)	7	8	GPIO3_17
GPIO0_4	9	10	NC
GPIO0_5	11	12	NC
GPIO0_6(UART3_RXD)	13	14	GPIO3_20
GND	15	16	GPIO3_21
UART0_TXD	17	18	GND
UART0_RXD	19	20	GPIO3_7(JTAG_EMU0)
I2C1_SCL(UART4_TXD)	21	22	GPIO3_8(JTAG_EMU1)
I2C1_SDA(UART4_RXD)	23	24	GPIO3_1
UART1_TXD	25	26	GPIO3_0(UART5_RXD)
UART1_RXD	27	28	GPIO3_2
UART1_CTSN	29	30	GPIO0_29(UART5_TXD)
UART1_RTSN	31	32	GPIO0_7(UART3_TXD)
GND	33	34	GND
USBHUB_DM3	35	36	NC
USBHUB_DP3	37	38	VDD_3v3
VDD_5v0	39	40	VDD_3v3

J12 커넥터에서 UART3 RX, TX, GND, VDD_3v3을 연결 했습니다.

보드 UART3 (J12) <-> USB to Serial UART 보드 <-> PC 연결 했습니다.



1.3.5. 소스 수정 및 확인

소프트 수정 작업은

"arch/arm/mach-omap2/board-am335xevm.c"

```
static struct pinmux_config uart3_pin_mux[] = {
    {"spi0_cs1.uart3_rxd", OMAP_MUX_MODE1 | AM33XX_PIN_INPUT_PULLUP},
    {"ecap0_in_pwm0_out.uart3_txd", OMAP_MUX_MODE1 | AM33XX_PULL_ENBL},
    {NULL, 0},
};
```

```
static void uart3_init(int evm_id, int profile)
{
```

```

MANGO_DBG_DEFAULT;

    setup_pin_mux(uart3_pin_mux);
    return;
}

```

```

static struct evm_dev_cfg evm_sk_dev_cfg[] = {
...
{uart3_init,          DEV_ON_BASEBOARD, PROFILE_ALL},

```

위와 같이 코드가 적용이 되어 있는지 확인 합니다.

"arch/arm/mach-omap2/mux33xx.c"

```

    _AM33XX_MUXENTRY(SPI0_CS1, 0,
        "spi0_cs1", "uart3_rxd", NULL, "mmc0_pow",
        NULL, "mmc0_sdcd", NULL, "gpio0_6"),
    _AM33XX_MUXENTRY(ECAP0_IN_PWM0_OUT, 0,
        "ecap0_in_pwm0_out", "uart3_txd", "spi1_cs1", NULL,
        "spi1_sclk", "mmc0_sdwp", NULL, "gpio0_7"),

```

"arch/arm/mach-omap2/omap_hwmod_33xx_data.c"

```

static struct omap_hwmod_dma_info uart3_edma_reqs[] = {
    { .name = "tx",    .dma_req = 30, },
    { .name = "rx",    .dma_req = 31, },
    { .dma_req = -1 }
};

static struct omap_hwmod_addr_space am33xx_uart3_addr_space[] = {
    {
        .pa_start = 0x48024000,
        .pa_end   = 0x48024000 + SZ_8K - 1,
        .flags    = ADDR_TYPE_RT,
    },
    {}
};

```

```

static struct omap_hwmod_ocp_if am33xx_l4_ls_uart3 = {
    .slave          = &am33xx_uart3_hwmod,
    .clk            = "uart3_ick",
    .addr           = am33xx_uart3_addr_space,
    .user           = OCP_USER_MPU,
};

static struct omap_hwmod_irq_info am33xx_uart3_irqs[] = {
    { .irq = 74 },
    { .irq = -1 }
};

static struct omap_hwmod_ocp_if *am33xx_uart3_slaves[] = {
    &am33xx_l4_ls_uart3,
};

static struct omap_hwmod am33xx_uart3_hwmod = {
    .name           = "uart3",
    .class          = &uart_class,
    .clkdm_name     = "l4ls_clkdm",
    .mpu_irqs       = am33xx_uart3_irqs,
    .main_clk       = "uart3_fck",
    .sdma_reqs      = uart3_edma_reqs,
    .prcm           = {
        .omap4       = {
            .clkctrl_offs = AM33XX_CM_PER_UART2_CLKCTRL_OFFSET,
            .modulemode  = MODULEMODE_SWCTRL,
        },
    },
    .slaves         = am33xx_uart3_slaves,
    .slaves_cnt     = ARRAY_SIZE(am33xx_uart3_slaves),
};

```

1.3.6. Uart3 테스트

UART3 채널 테스트

```
# stty -F /dev/ttyO3 115200
```

Baudrate : 115200 설정합니다.

TX 테스트

```
# ls -al > /dev/ttyO3
```

위와 같이 명령을 입력하면 다른 터미널 창에서 출력이 됩니다.

```
total 144
drwxrwxr-x  5 1001 1001  4096 Apr 20  2015 ./
drwxrwxr-x 18 root root  4096 Aug 10  2015 ../
-rw-rw-r--  1 root root    0 Nov 30  2013 .bash_history
-rw-rw-r--  1 root root  175 Nov 30  2013 .bash_logout
-rw-rw-r--  1 root root  161 Nov 30  2013 .bash_profile
-rwxr--r--  1 1001 1001  1019 Apr 20  2015 S99_total_mp_process.sh*
drwxrwxr-x  2 1001 1001  4096 Feb 24  2015 boot_dir/
-rwxr--r--  1 1001 1001   150 Feb 24  2015 flash_eraseall*
-rw-rw-r--  1 1001 1001 73805 May 19  2015 g_mass_storage.ko
drwxrwxr-x  2 1001 1001  4096 Feb  5  2014 nfs_mount/
-rwxr-xr-x  1 1001 1001   247 Jan 16 17:16 nfs_mount.sh*
-rwxrwxr-x  1 1001 1001 10391 Mar 21  2014 serial_test*
-rwxr-xr-x  1 1001 1001   196 Feb 24  2015 tmp_kernel_update.sh*
-rwxr-xr-x  1 1001 1001   344 Mar  9  2015 tmp_uboot_update.sh*
drwxr-xr-x  2 1001 1001  4096 Feb 25  2015 total_mp_process/
-rwxr-xr-x  1 1001 1001   154 Apr 23  2014 usb_dev_mass.sh*
```

RX 테스트

디버깅 터미널 창에서 아래와 같이 입력하고, UART3가 연결 된 터미널 창에서 아래와 같이 입력하면 디버깅 터미널 창으로 출력이 됩니다.

```
[root@localhost ~]# cat /dev/ttyO3
```

```
Mango UART3 Test OK
```

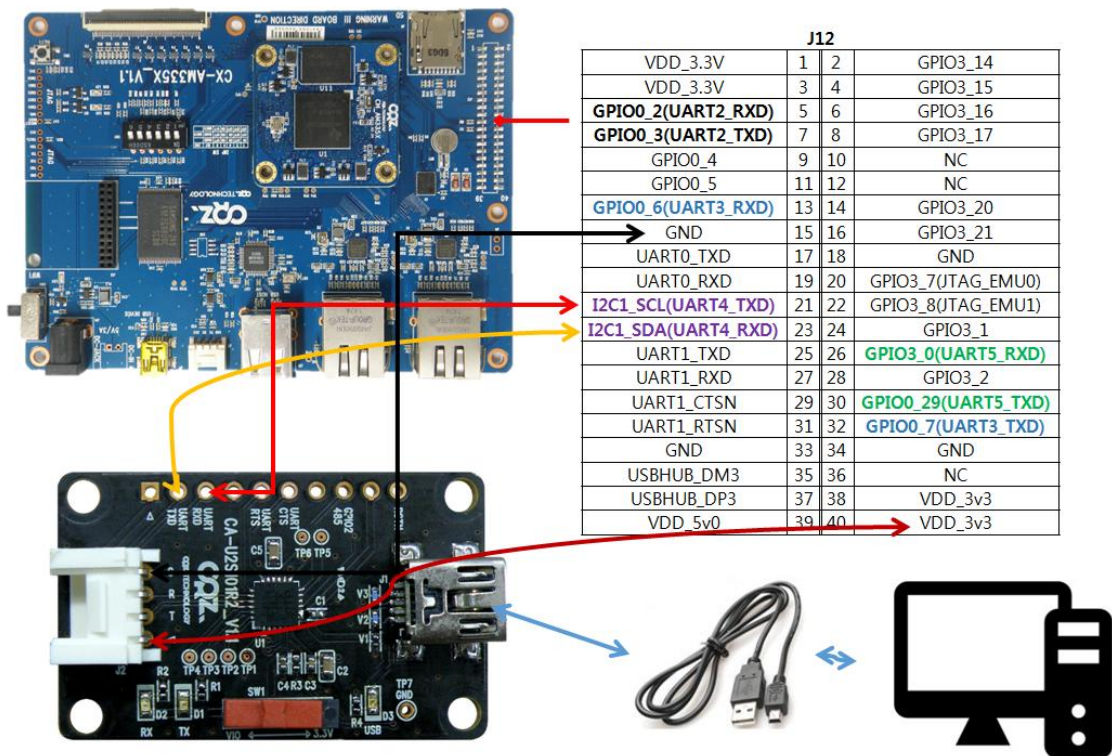
1.4. Mango-AM335x-ST 보드 UART4검증

UART4 사용하려면, J12 핀 커넥터에서

21번 : UART4_TXD

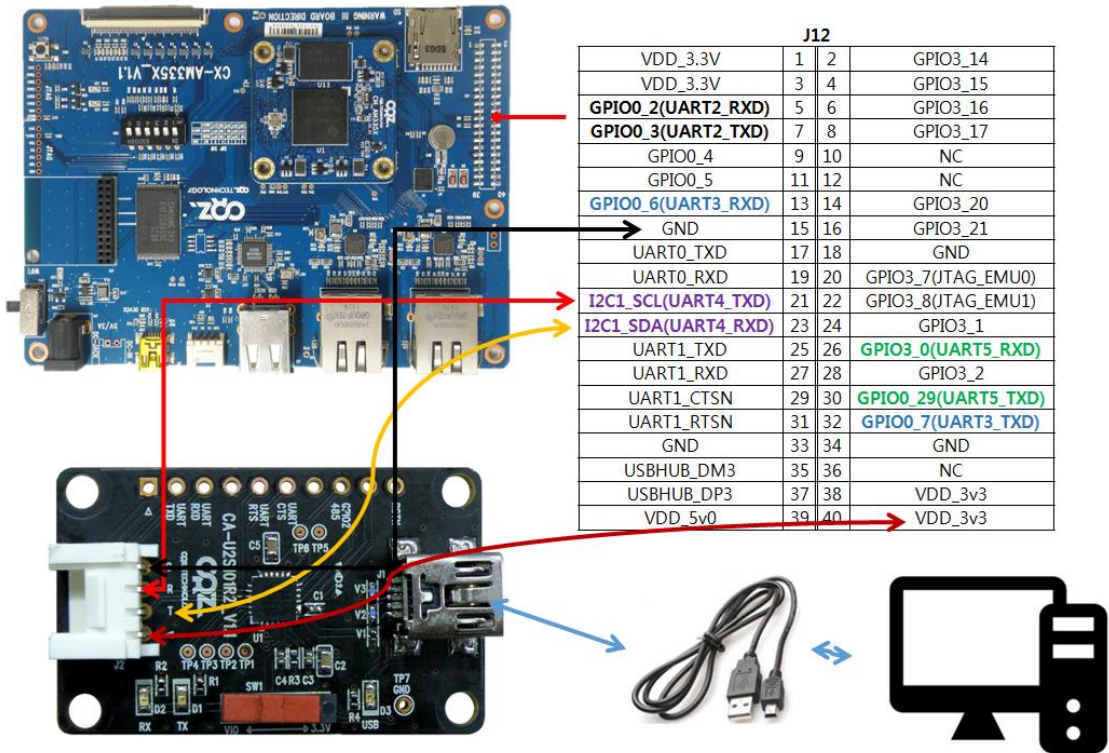
23번 : UART4_RXD

연결합니다.



위와 같이 연결을 합니다.

4pin cable 아래와 같이 연결해도 됩니다.



1.4.7. 소스 수정

<http://www.ti.com/lit/ds/symlink/am3358.pdf>

에 보면

UART0_RTSn	uart0_rtsn	0	O
	uart4_txd	1	O
	dcan1_rx	2	I
	I2C1_SCL	3	I/OD
	spi1_d1	4	I/O
	spi1_cs0	5	I/O
	pr1_edc_sync1_out	6	O
	gpio1_9	7	I/O

GPIO1_9에 uart4_txd 로 mux 설정하도록 되어 있습니다.

UART1_RXD는 GPIO1_8, Mode 1로 설정하면 되도록 되어 있습니다.

UART0_CTSn	uart0_ctsn	0	I
	uart4_rxd	1	I
	dcan1_tx	2	O
	I2C1_SDA	3	I/OD
	spi1_d0	4	I/O
	timer7	5	I/O
	pr1_edc_sync0_out	6	O
	gpio1_8	7	I/O

"arch/arm/mach-omap2/mux33xx.c" 수정

```

_AM33XX_MUXENTRY(UART0_CTSN, 0,
    "uart0_ctsn", "uart4_rxd", "d_can1_tx", "i2c1_sda",
    "spi1_d0", NULL, NULL, "gpio1_8"),
_AM33XX_MUXENTRY(UART0_RTSN, 0,
    "uart0_rtsn", "uart4_txd", "d_can1_rx", "i2c1_scl",
    "spi1_d1", "spi1_cs0", NULL, "gpio1_9"),

```

"arch/arm/mach-omap2/board-am335xevm.c" 수정

I2C 1번 채널을 사용할 경우에는 막도록 하기 바랍니다. 둘 중에 하나만 사용이 가능합니다.

"arch/arm/mach-omap2/board-am335xevm.c"

```
//      {i2c1_init,          DEV_ON_BASEBOARD, PROFILE_ALL},
```

```
static struct pinmux_config uart4_pin_mux[] = {
    {"uart0_ctsn.uart4_rxd", OMAP_MUX_MODE1 | AM33XX_PIN_INPUT_PULLUP},
    {"uart0_rtsn.uart4_txd", OMAP_MUX_MODE1 | AM33XX_PULL_ENBL},
    {NULL, 0},
};
```

```
static void uart4_init(int evm_id,int profile)
{
    MANGO_DBG_DEFAULT;

    setup_pin_mux(uart4_pin_mux);
    return;
}
```



```
}
```

```
static struct evm_dev_cfg evm_sk_dev_cfg[] = {  
...  
{uart4_init,          DEV_ON_BASEBOARD, PROFILE_ALL},//crazyboys 20150812
```

“arch/arm/mach-omap2/omap_hwmod_33xx_data.c”

1.4.8. UART4 테스트

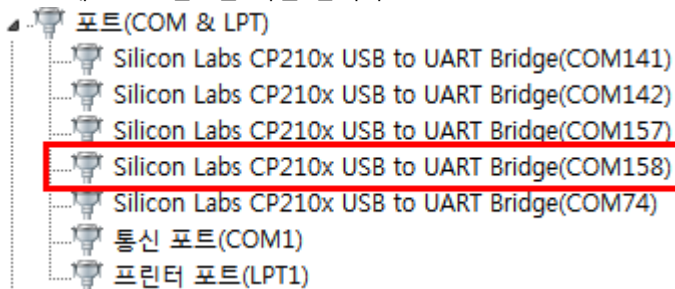
Uart4 TX, RX 테스트를 진행합니다.

UART0 디버깅 터미널 프로그램을 실행을 합니다.
아래와 같이 명령을 입력합니다.

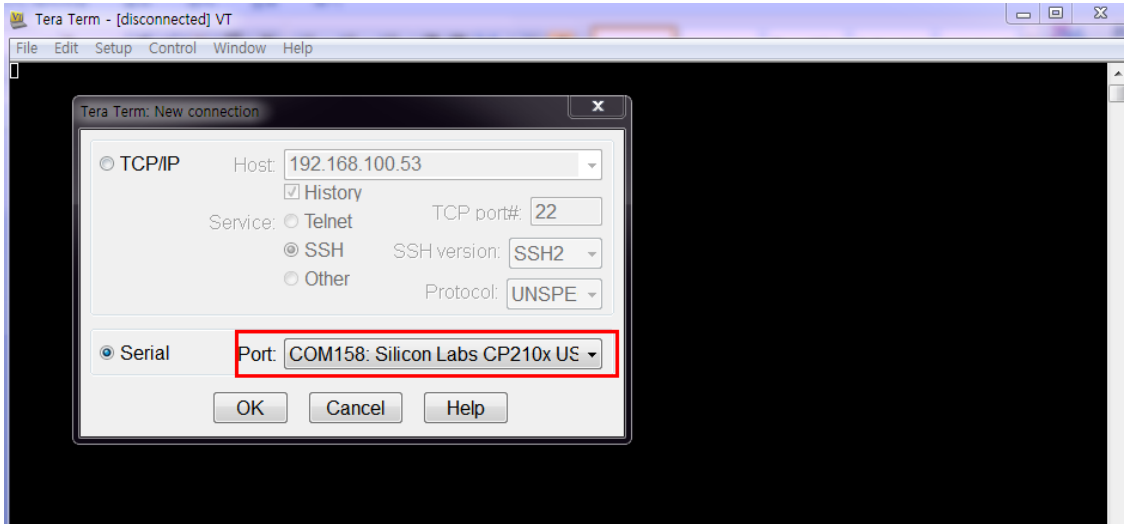
```
[root@localhost ~]# stty -F /dev/ttyO4 115200  
[root@localhost ~]# stty -F /dev/ttyO4  
speed 115200 baud; line = 0;  
-brkint -imaxbel
```

Baudrate를 115200으로 변경합니다.

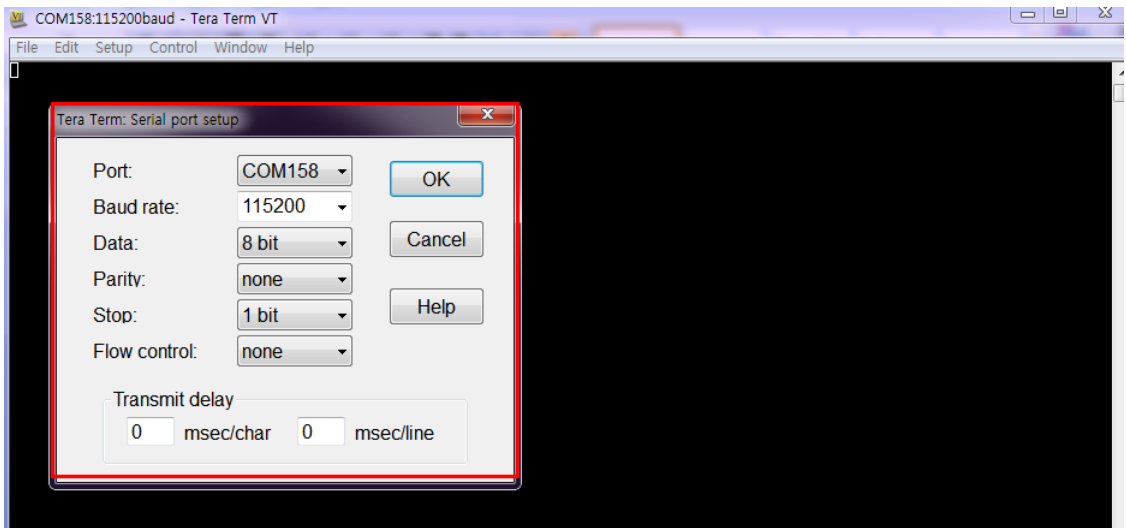
UART4에 COM 번호를 확인 합니다.



터미널 프로그램 중 데라텀을 실행합니다.



설정은 아래와 같이 입력합니다.



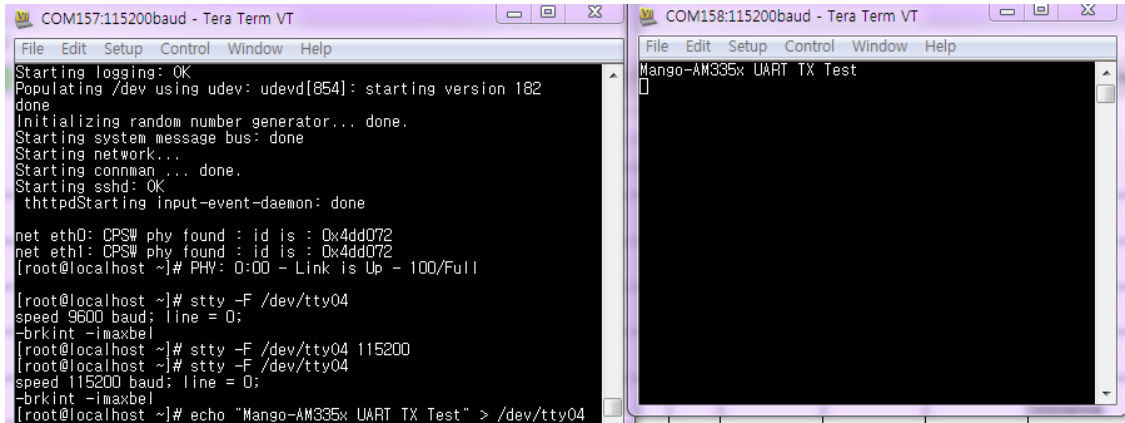
TX 테스트

UART0 디버깅 터미널 창에서 아래와 같이 입력을 하면,
 UART4 연결한 터미널 창에서 입력한 값이 출력이 됩니다.

```
# echo "Mango-AM335x UART TX Test" > /dev/ttyO4
```

아래 왼쪽 : UART0 디버깅 터미널

아래 오른쪽 : UART4 터미널 창

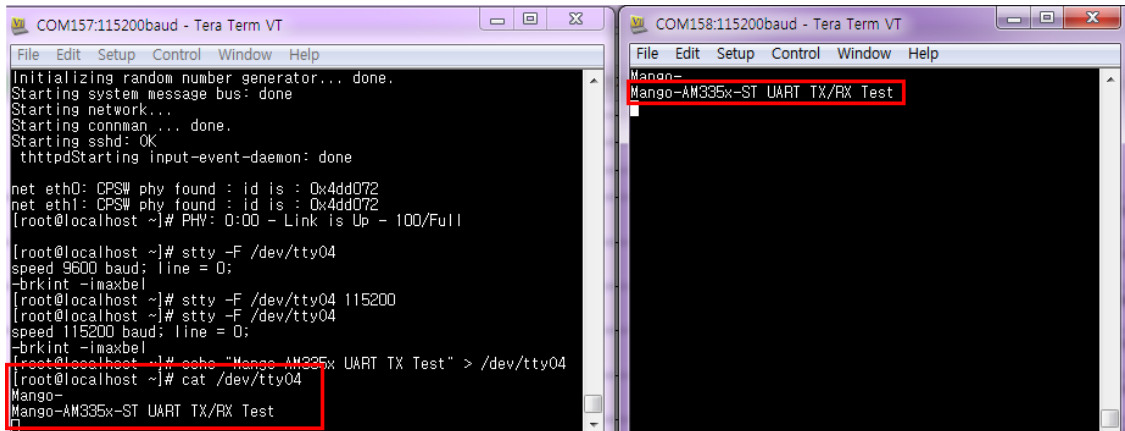


RX 테스트

UART0 디버깅 터미널 창에서 Receive를 하도록 명령을 입력합니다.

```
# cat /dev/ttyO4
```

UART4 터미널 창으로 커서를 이동한 후 아래와 같이 입력하고, 엔터키를 치면
UART0 디버깅 터미널 창으로 입력한 글자가 출력이 됩니다.

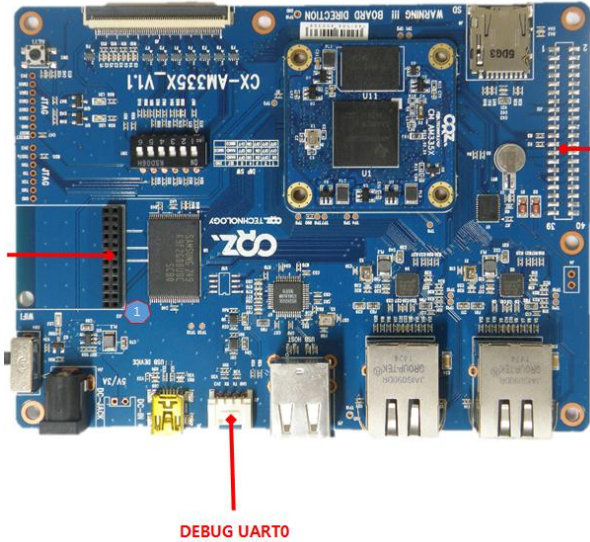


1.5. Mango-AM335x-ST 보드 UART5검증

J12 에서

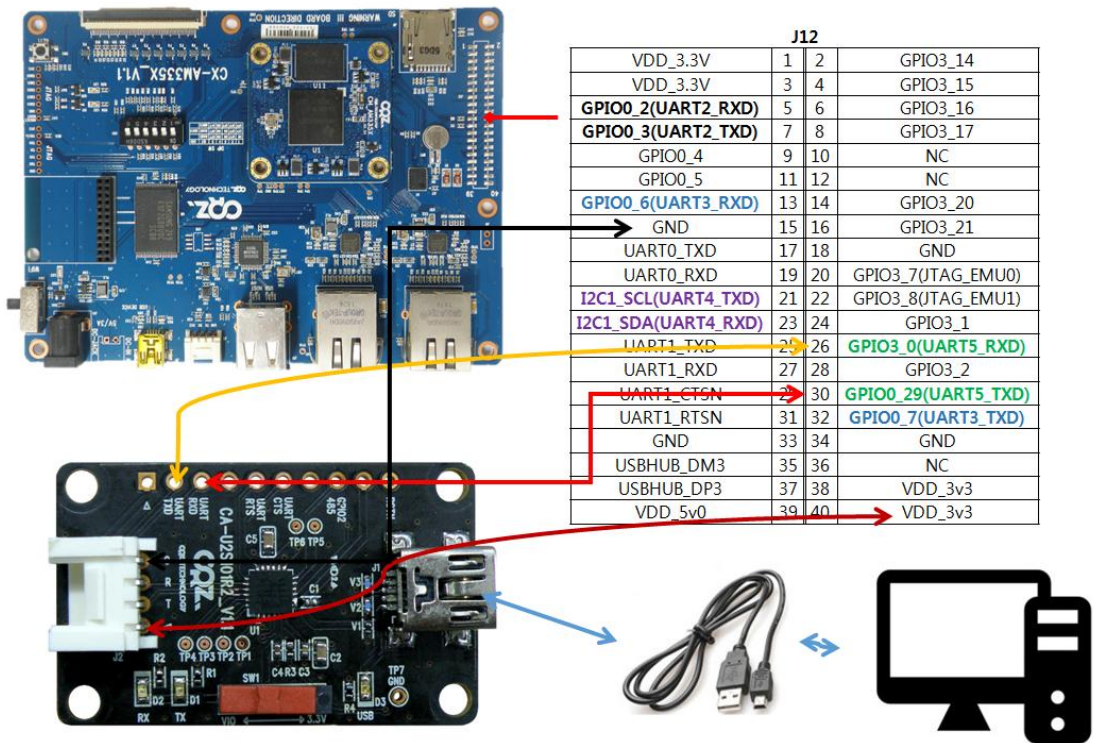
26번 : GPIO3_0 (UART5_RXD)

30번 : GPIO0_29 (UART5_TXD)
 핀이 나와 있습니다.



J12			
VDD_3.3V	1	2	GPIO3_14
VDD_3.3V	3	4	GPIO3_15
GPIO0_2(UART2_RXD)	5	6	GPIO3_16
GPIO0_3(UART2_TXD)	7	8	GPIO3_17
GPIO0_4	9	10	NC
GPIO0_5	11	12	NC
GPIO0_6(UART3_RXD)	13	14	GPIO3_20
GND	15	16	GPIO3_21
UART0_TXD	17	18	GND
UART0_RXD	19	20	GPIO3_7(ITAG_EMU0)
I2C1_SCL(UART4_TXD)	21	22	GPIO3_8(ITAG_EMU1)
I2C1_SDA(UART4_RXD)	23	24	GPIO3_1
UART1_TXD	25	26	GPIO3_0(UART5_RXD)
UART1_RXD	27	28	GPIO3_2
UART1_CTSN	29	30	GPIO0_29(UART5_TXD)
UART1_RTSN	31	32	GPIO0_7(UART3_TXD)
GND	33	34	GND
USBHUB_DM3	35	36	NC
USBHUB_DP3	37	38	VDD_3v3
VDD_5v0	39	40	VDD_3v3

Mango-AM335x-ST 보드 J12 에서 26, 30번 , VDD_3v3, GND 를 USB to Serial UART 모듈과 연결하고, mini USB cable 을 PC에 연결을 합니다.
 VDD_3v3은 꼭 연결하지 않아도 됩니다.



1.5.9. Uart5 검증을 위한 소스 수정

소프트웨어 수정 작업

<http://www.ti.com/lit/ds/symlink/am3358.pdf> 를 살펴 보면 아래와 같이 MODE 3에 uart5_rxd가 되어 있습니다.

MII1_COL	gmii1_col	0	I
	rmii2_refclk	1	I/O
	spi1_sclk	2	I/O
	uart5_rxd	3	I
	mcasp1_axr2	4	I/O
	mmc2_dat3	5	I/O
	mcasp0_axr2	6	I/O
	gpio3_0	7	I/O

"arch/arm/mach-omap2/mux33xx.c"

```
_AM33XX_MUXENTRY(MII1_COL, 0,  
    "mii1_col", "rmii2_refclk", "spi1_sclk", "uart5_rxd",  
    "mcasep1_axr2", "mmc2_dat3", "mcasep0_axr2", "gpio3_0"),
```

Uart5_rxd를 mux mode에 정의를 합니다.

<http://www.ti.com/lit/ds/symlink/am3358.pdf> 를 살펴 보면 아래와 같이 MODE 3에
uart5_txd가 되어 있습니다.

RMII1_REF_CLK	rmii1_refclk	0	I/O
	xdma_event_intr2	1	I
	spi1_cs0	2	I/O
	uart5_txd	3	O
	mcasep1_axr3	4	I/O
	mmc0_pow	5	O
	mcasep1_ahclkx	6	I/O
	gpio0_29	7	I/O

"arch/arm/mach-omap2/mux33xx.c"

```
_AM33XX_MUXENTRY(MII1_REFCLK, 0,  
    "rmii1_refclk", NULL, "spi1_cs0", "uart5_txd",  
    "mcasep1_axr3", "mmc0_pow", "mcasep1_ahclkx", "gpio0_29"),
```

수정을 합니다.

Pin mux를 지정합니다.

"arch/arm/mach-omap2/board-am335xevm.c"

```
static struct pinmux_config uart5_pin_mux[] = {  
    {"mii1_col.uart5_rxd", OMAP_MUX_MODE3 | AM33XX_PIN_INPUT_PULLUP},  
    {"rmii1_refclk.uart5_txd", OMAP_MUX_MODE3 | AM33XX_PULL_ENBL},  
    {NULL, 0},  
};
```

```

static void uart5_init(int evm_id,int profile)
{
    MANGO_DBG_DEFAULT;

    setup_pin_mux(uart5_pin_mux);
    return;
}

```

```

static struct evm_dev_cfg evm_sk_dev_cfg[] = {
...
{uart5_init,          DEV_ON_BASEBOARD, PROFILE_ALL},//crazyboys 20150812

```

"arch/arm/mach-omap2/omap_hwmod_33xx_data.c" 파일 수정

```

static struct omap_hwmod_addr_space am33xx_uart5_addr_space[] = {
    {
        .pa_start = 0x481A8000,
        .pa_end   = 0x481A8000 + SZ_8K - 1,
        .flags    = ADDR_TYPE_RT,
    },
    {}
};

static struct omap_hwmod_ocp_if am33xx_l4_ls_uart5 = {
    .slave      = &am33xx_uart5_hwmod,
    .clk        = "uart5_ick",
    .addr       = am33xx_uart5_addr_space,
    .user       = OCP_USER_MPU,
};

static struct omap_hwmod_irq_info am33xx_uart5_irqs[] = {
    { .irq = 45 },
    { .irq = -1 }
};

static struct omap_hwmod_ocp_if *am33xx_uart5_slaves[] = {

```



```

        &am33xx_l4_ls_uart5,
};

static struct omap_hwmod am33xx_uart5_hwmod = {
    .name          = "uart5",
    .class         = &uart_class,
    .clkdm_name   = "l4ls_clkdm",
    .mpu_irqs     = am33xx_uart5_irqs,
    .main_clk     = "uart5_fck",
    .sdma_reqs    = uart1_edma_reqs,
    .prcm         = {
        .omap4     = {
            .clkctrl_offs = AM33XX_CM_PER_UART4_CLKCTRL_OFFSET,
            .modulemode  = MODULEMODE_SWCTRL,
        },
    },
    .slaves       = am33xx_uart5_slaves,
    .slaves_cnt   = ARRAY_SIZE(am33xx_uart5_slaves),
};

```

커널 컴파일을 합니다. 커널 이미지를 write를 합니다.

1.5.10. UART5 TX , RX 테스트

Uart5 TX, RX 테스트를 진행합니다.

UART0 디버깅 터미널 프로그램을 실행을 합니다.

아래와 같이 명령을 입력합니다.

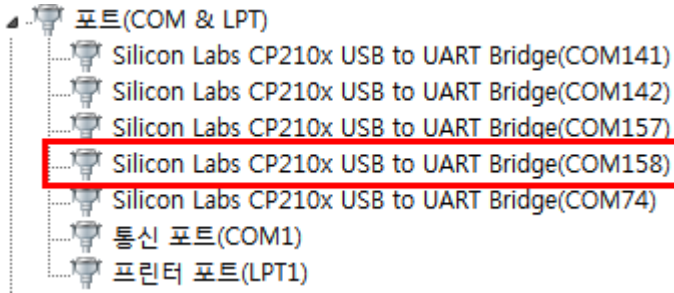
```

[root@localhost ~]# stty -F /dev/ttyO5 115200
[root@localhost ~]# stty -F /dev/ttyO5
speed 115200 baud; line = 0;
-brkint -imaxbel

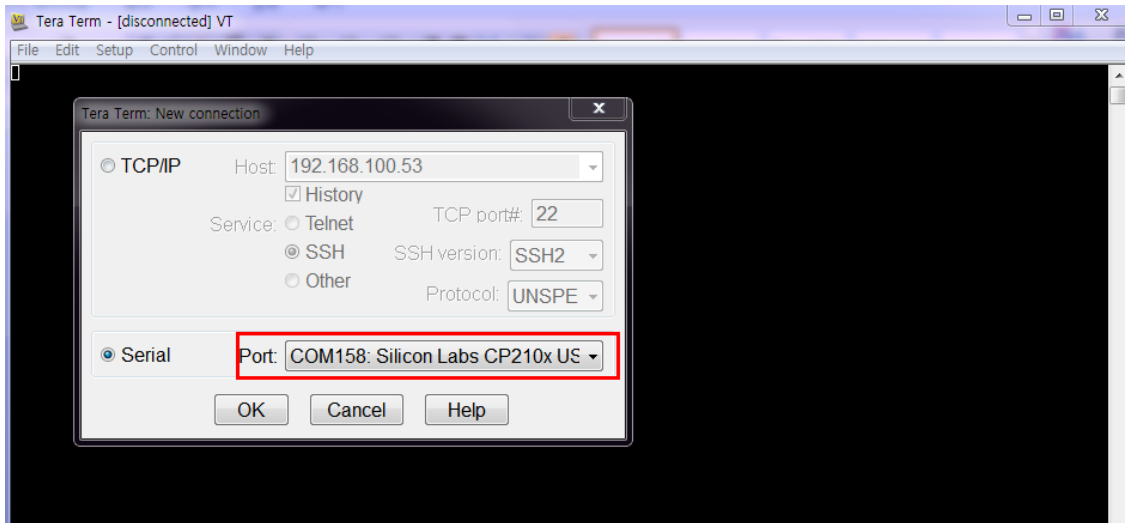
```

Baudrate를 115200으로 변경합니다.

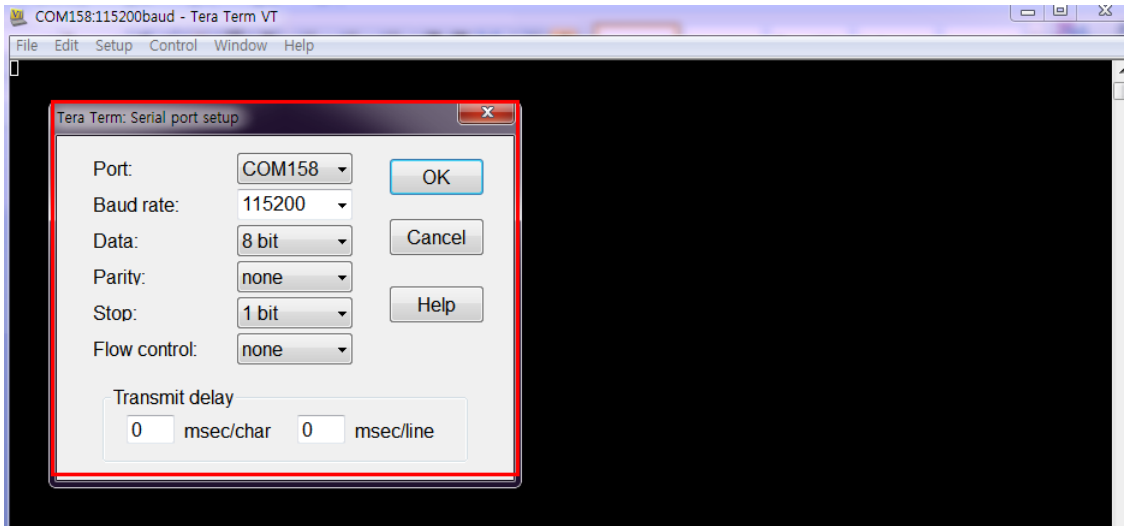
UART5에 COM 번호를 확인 합니다.



터미널 프로그램 중 테라텀을 실행합니다.



설정은 아래와 같이 입력합니다.



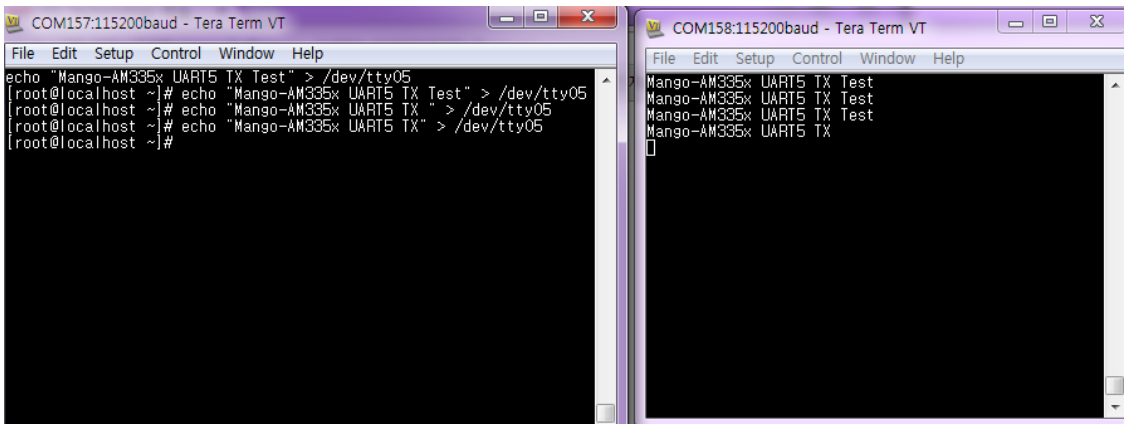
TX 테스트

UART0 디버깅 터미널 창에서 아래와 같이 입력을 하면,
UART5 연결한 터미널 창에서 입력한 값이 출력이 됩니다.

```
# echo "Mango-AM335x UART TX Test" > /dev/ttyO5
```

아래 왼쪽 : UART0 디버깅 터미널

아래 오른쪽 : UART5 터미널 창



RX 테스트

UART0 디버깅 터미널 창에서 Receive를 하도록 명령을 입력합니다.

```
# cat /dev/ttyO5
```

UART5 터미널 창으로 커서를 이동한 후 아래와 같이 입력하고 , 엔터키를 치면
UART0 디버깅 터미널 창으로 입력한 글자가 출력이 됩니다.

